Digital Watermarking

Presented by Melinos Averkiou

History

- 1282 – Paper Watermarks
- 1779 – Counterfeiting
- 1954 – Watermarking music
- 1988 – First use of the term Digital Watermark
- End of 1990s – large interest in watermarking

Applications

- Broadcast monitoring
- Owner identification
- Transaction Tracking
- Content Authentication
- Copy Control
- ..many more

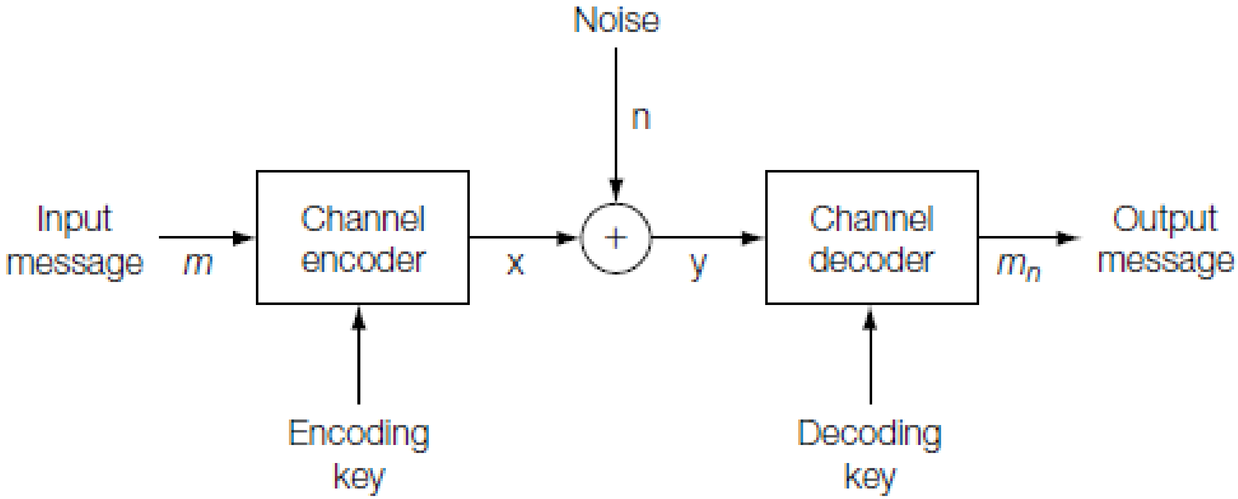
Watermarking Properties

- Embedding effectiveness
- Fidelity
- Payload
- Blind or informed detection
- False positive rate
- Robustness

Watermarking models

1. Communication-Based

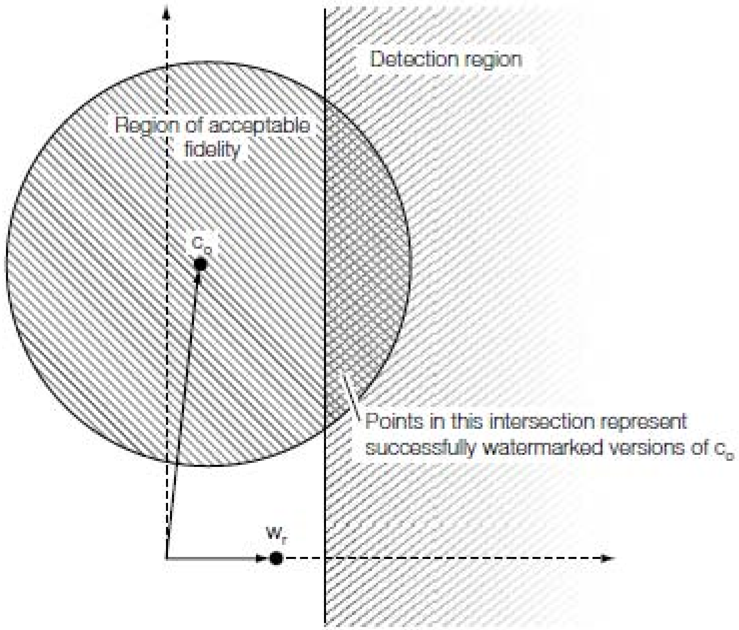
- Without side-information
- With side-information



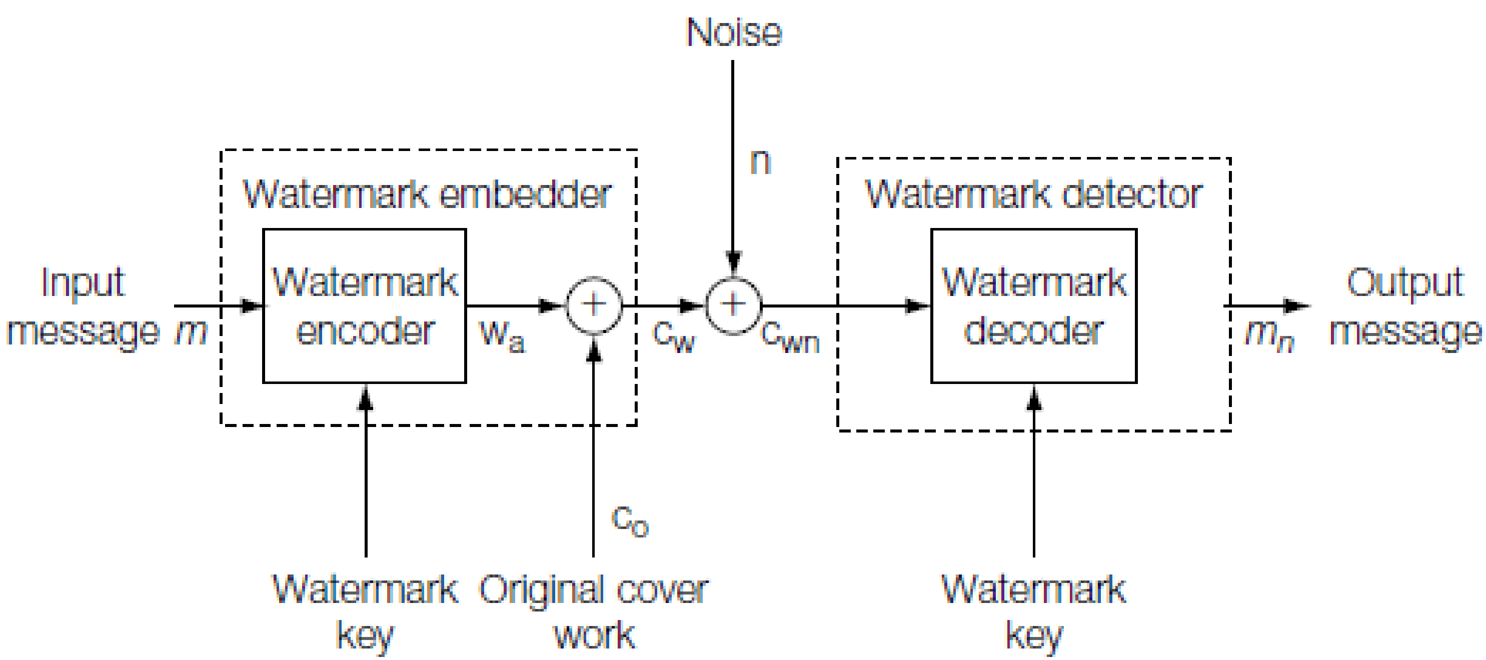
Watermarking Models

2. Geometric

- Media Space
  - Embedding Region
  - Detection Region
  - Region of acceptable fidelity
- Marking Space



Watermarking without side-information



Blind Embedding and Linear Correlation Detection

Embedder:

1. Choose one random reference pattern(w_r)
2. Choose message mark for 1 and 0

   α controls the embedding strength

$$\mathbf{w}_m = \begin{cases} \mathbf{w_r} & \text{if } m = 1 \\ -\mathbf{w_r} & \text{if } m = 0 \end{cases}$$

$$\mathbf{w_a} = \alpha \mathbf{w}_m$$

$$\mathbf{c_w} = \mathbf{c_o} + \mathbf{w_a}.$$

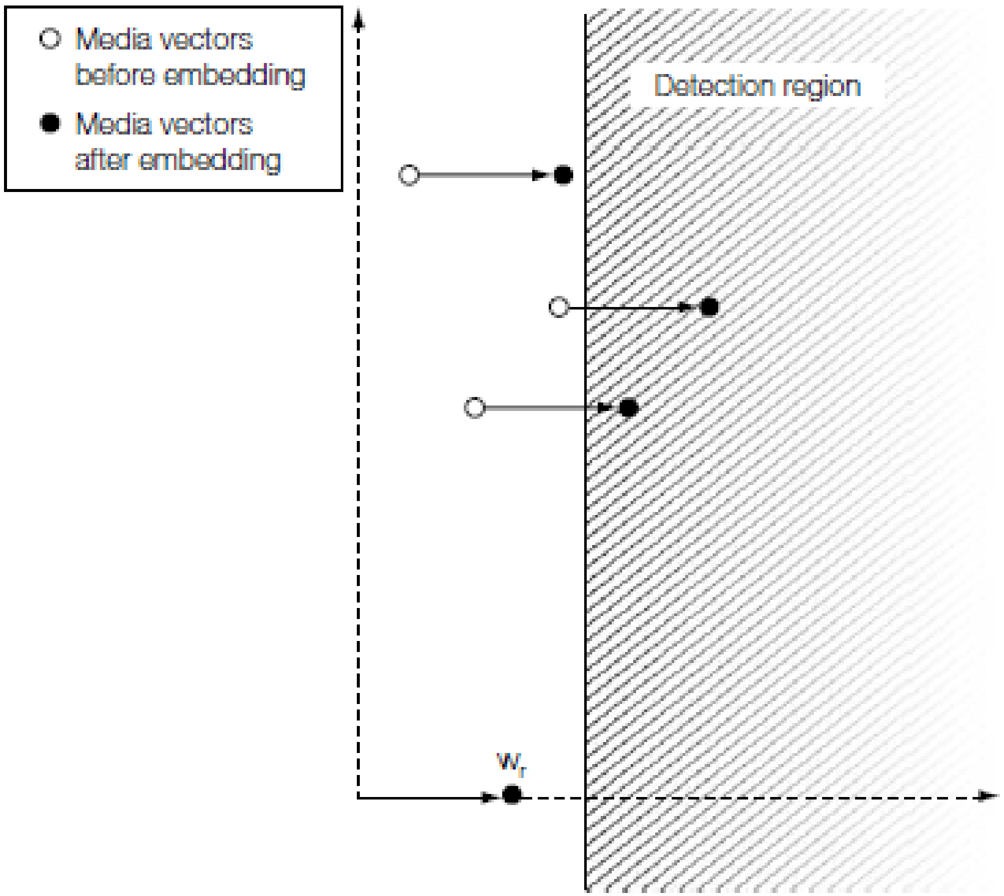
Detector:

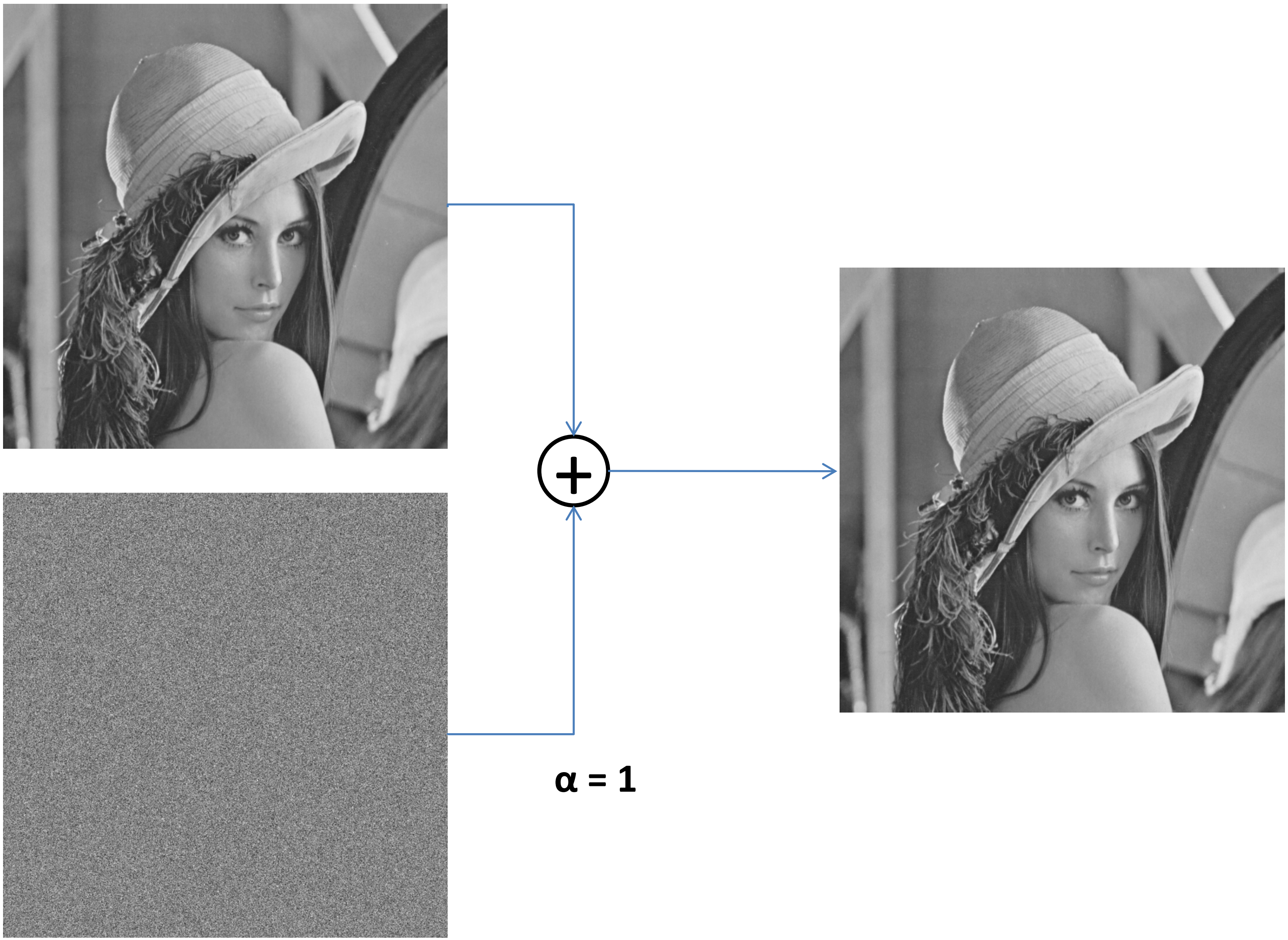
1. Calculate linear correlation z_{lc}
2. Detect message according to z_{lc}

$$z_{lc}(\mathbf{c}, \mathbf{w_r}) = \frac{1}{N}\mathbf{c} \cdot \mathbf{w_r} = \frac{1}{N}\sum_{x,y} \mathbf{c}[x,y]\,\mathbf{w_r}[x,y],$$

$$m_n = \begin{cases} 1 & \text{if } z_{lc}(\mathbf{c}, \mathbf{w_r}) > \tau_{lc} \\ \textit{no watermark} & \text{if } -\tau_{lc} \le z_{lc}(\mathbf{c}, \mathbf{w_r}) \le \tau_{lc} \\ 0 & \text{if } z_{lc}(\mathbf{c}, \mathbf{w_r}) < -\tau_{lc}. \end{cases}$$

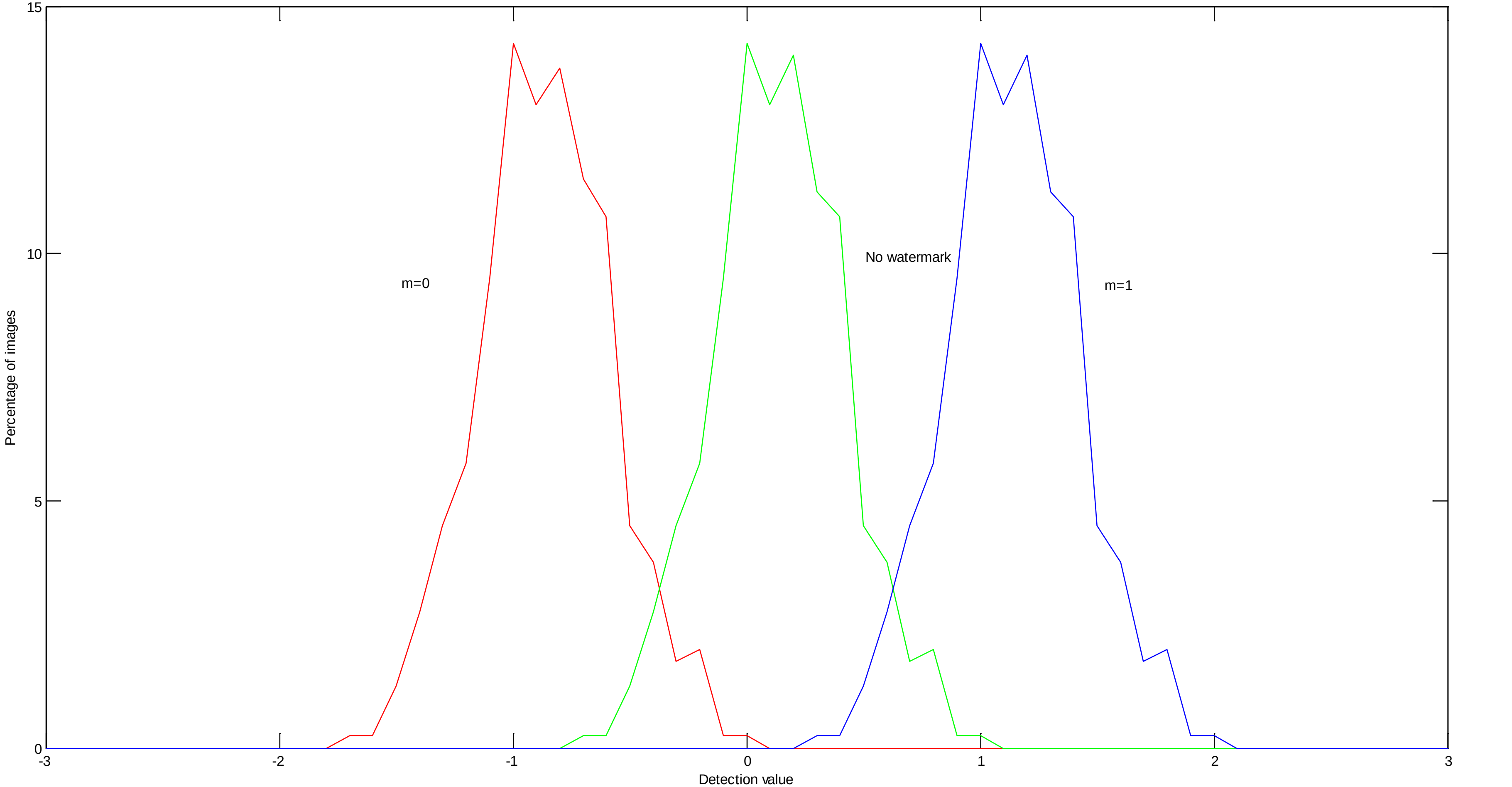
Geometric Interpretation




α = 1

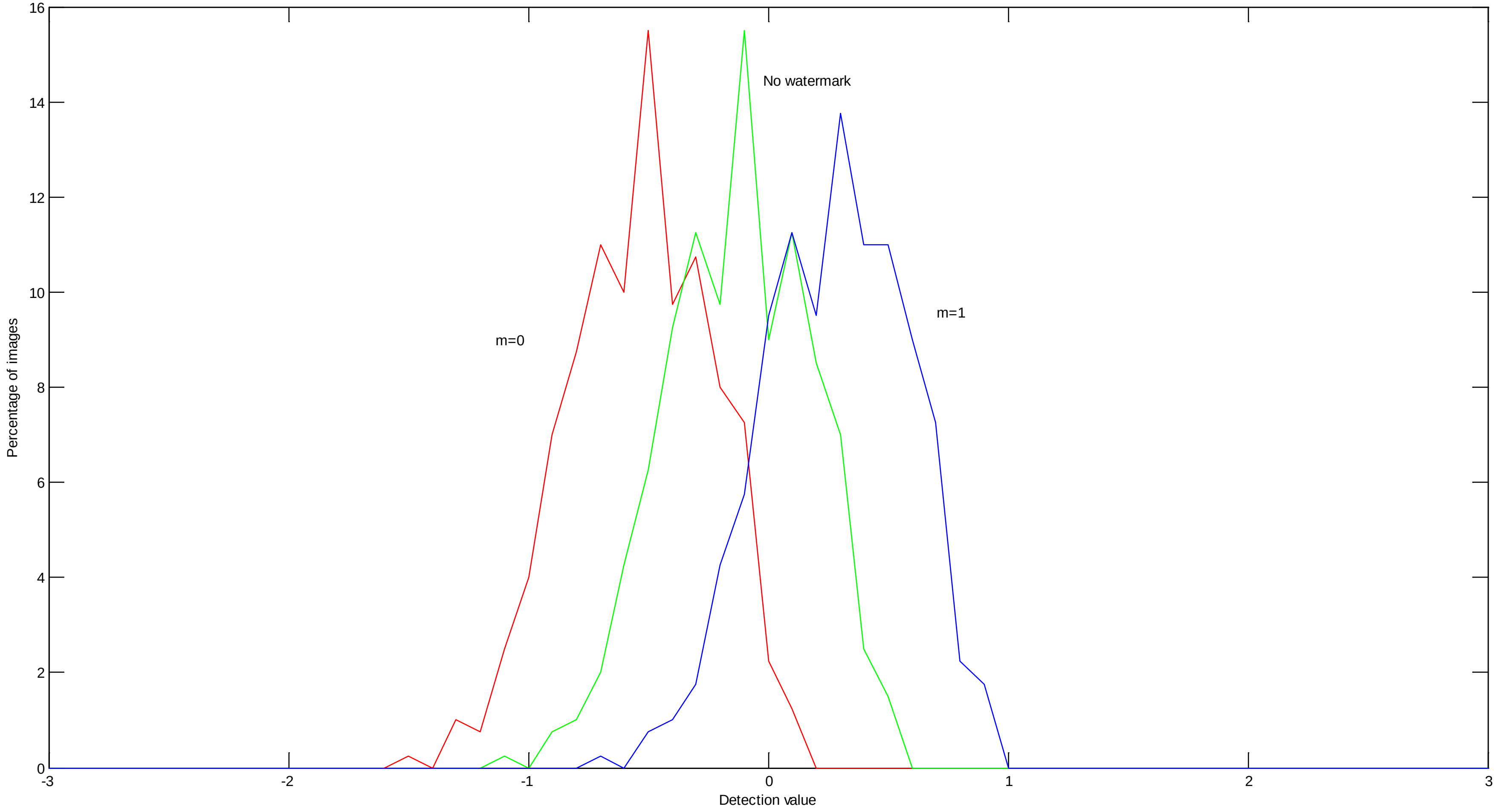
Effectiveness

400 images (112 x 92 pixels)



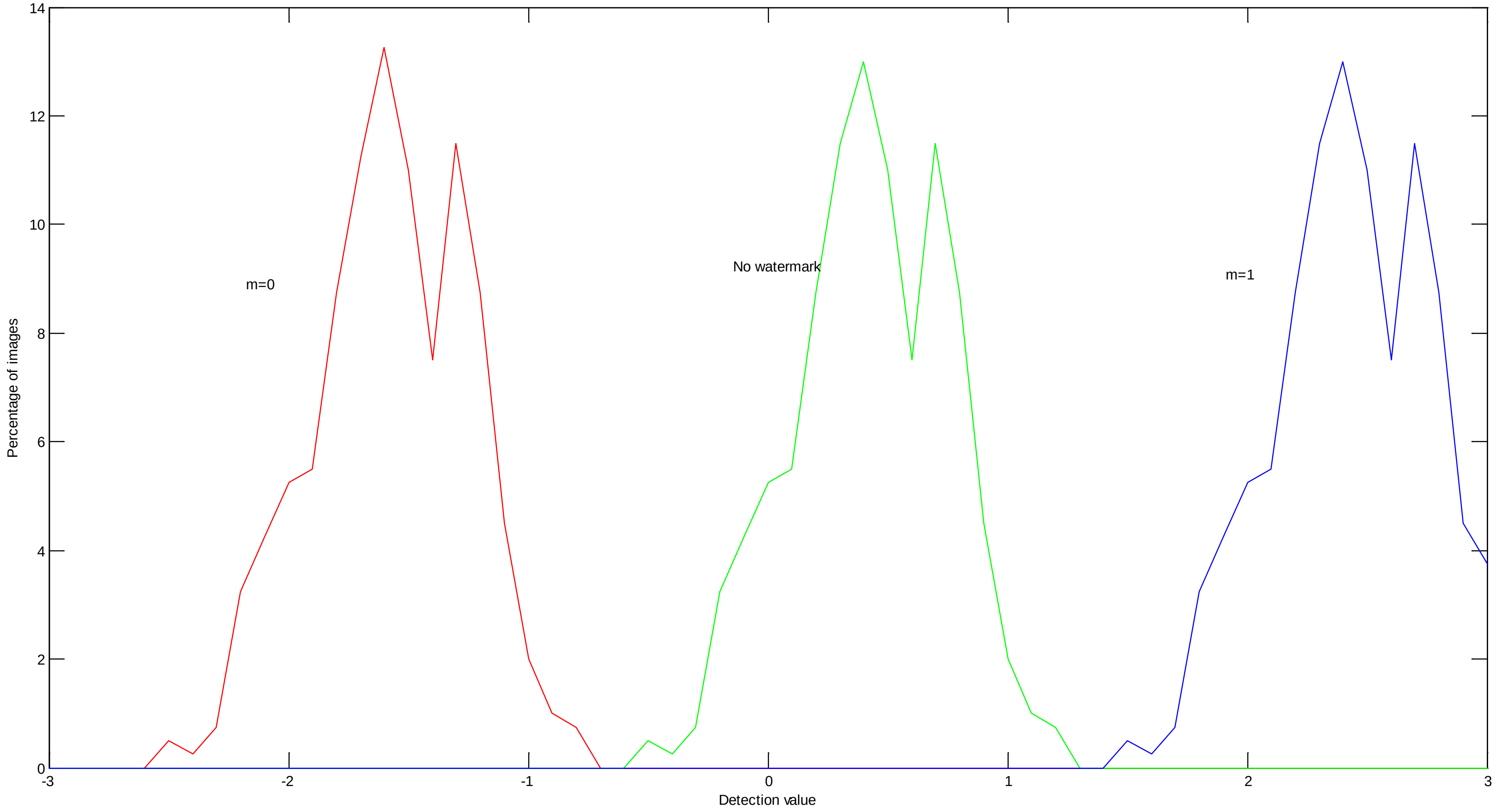
Reference pattern is very important

Low pass reference pattern

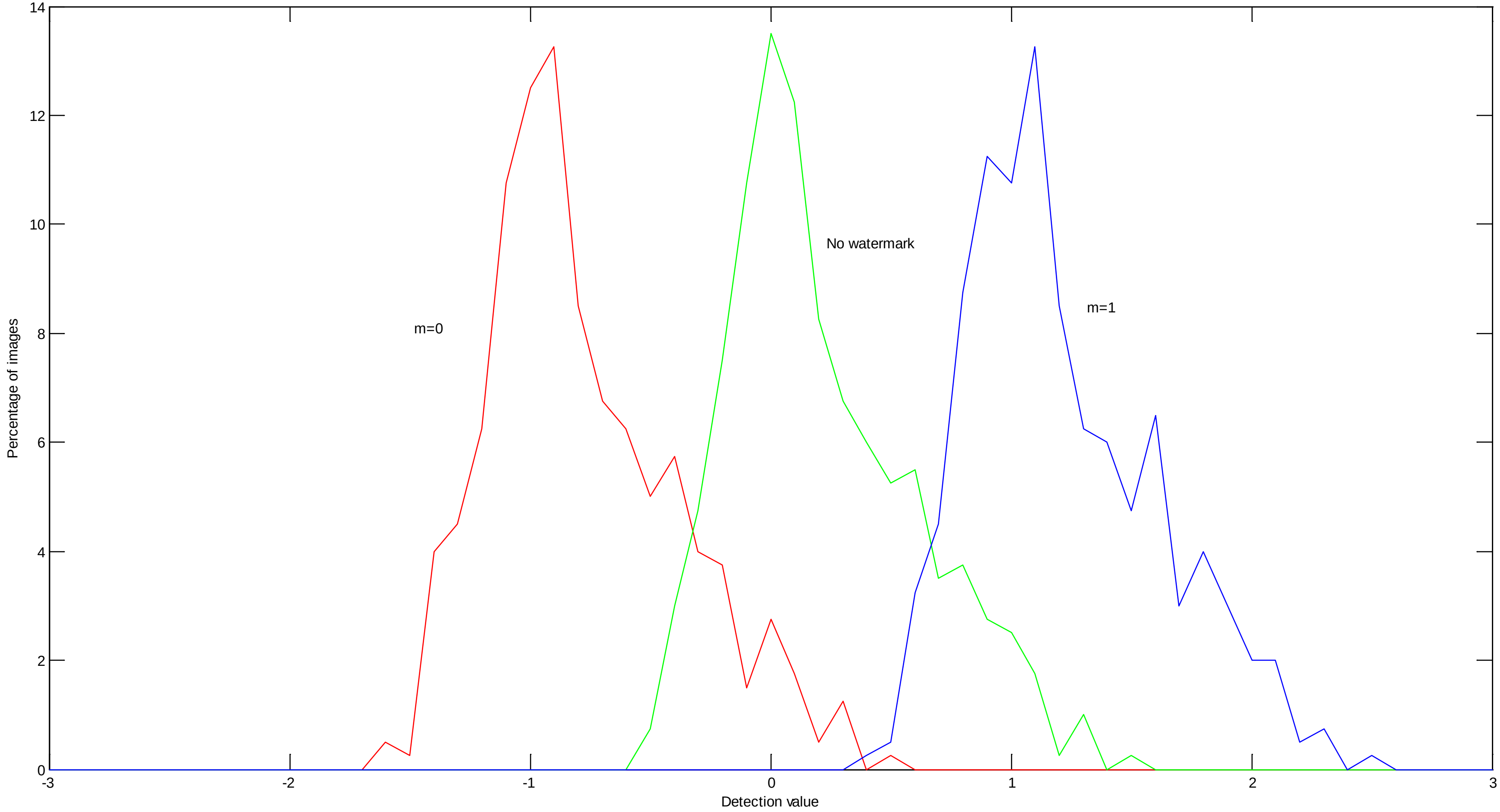


α is very important

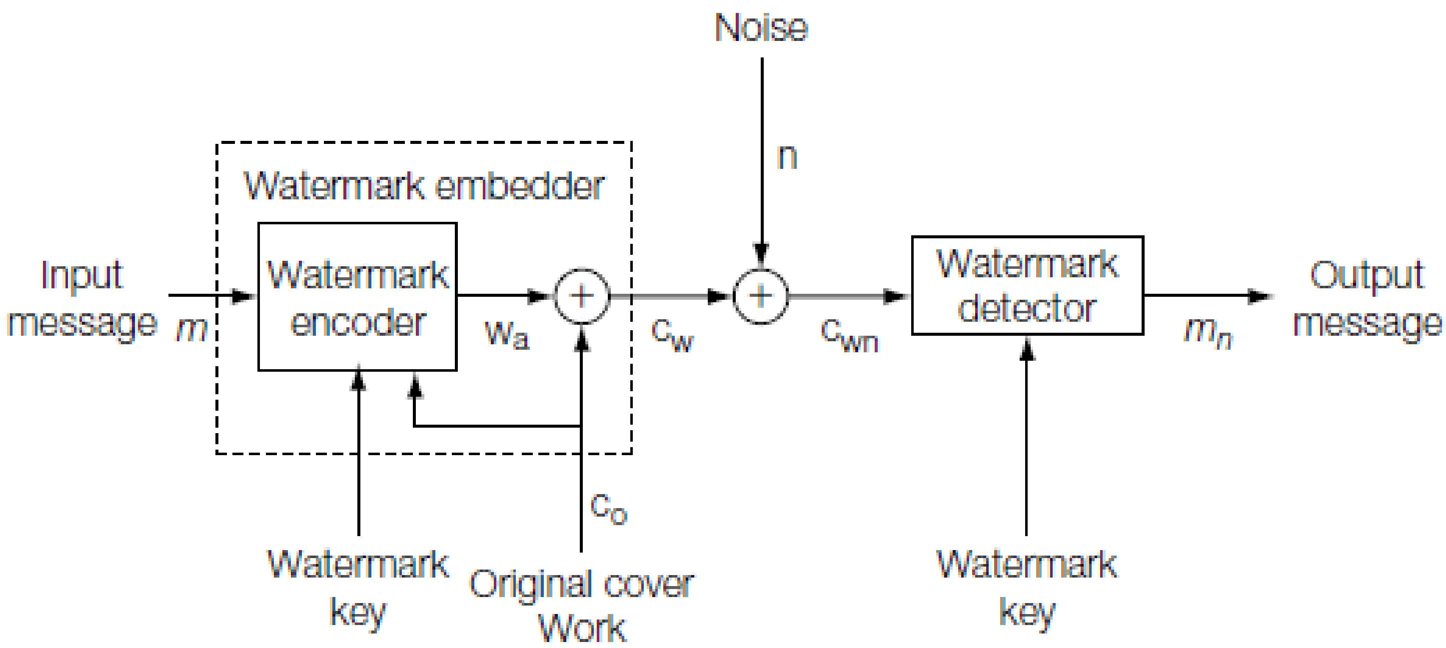
$\alpha = 2$



Adding noise



Watermarking with side-information



Informed Embedding and Linear Correlation Detection

Embedder:

1. Choose one random reference pattern(w_r)
2. Calculate α so that we have 100% effectiveness
3. Choose message mark for 1 and 0

$$\mathbf{w}_m = \begin{cases} \mathbf{w_r} & \text{if } m = 1 \\ -\mathbf{w_r} & \text{if } m = 0 \end{cases}$$

$$\mathbf{w_a} = \alpha \mathbf{w}_m$$

$$\mathbf{c_w} = \mathbf{c_o} + \mathbf{w_a}.$$

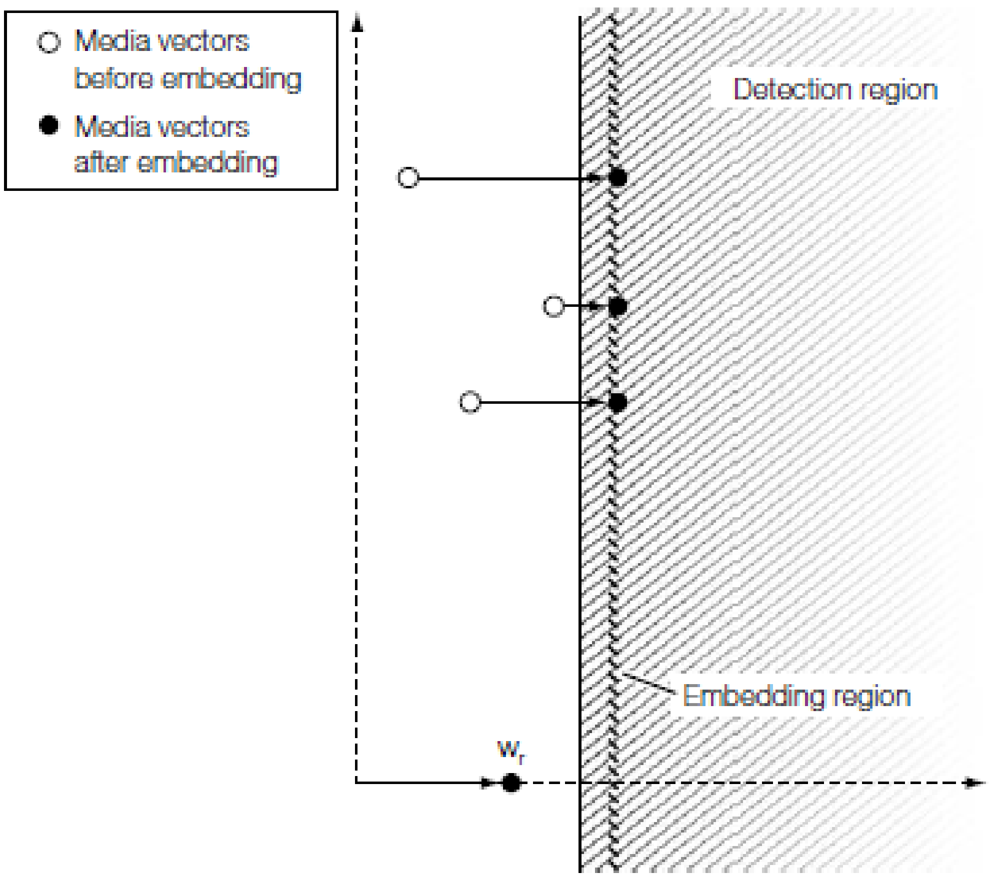
Detector:

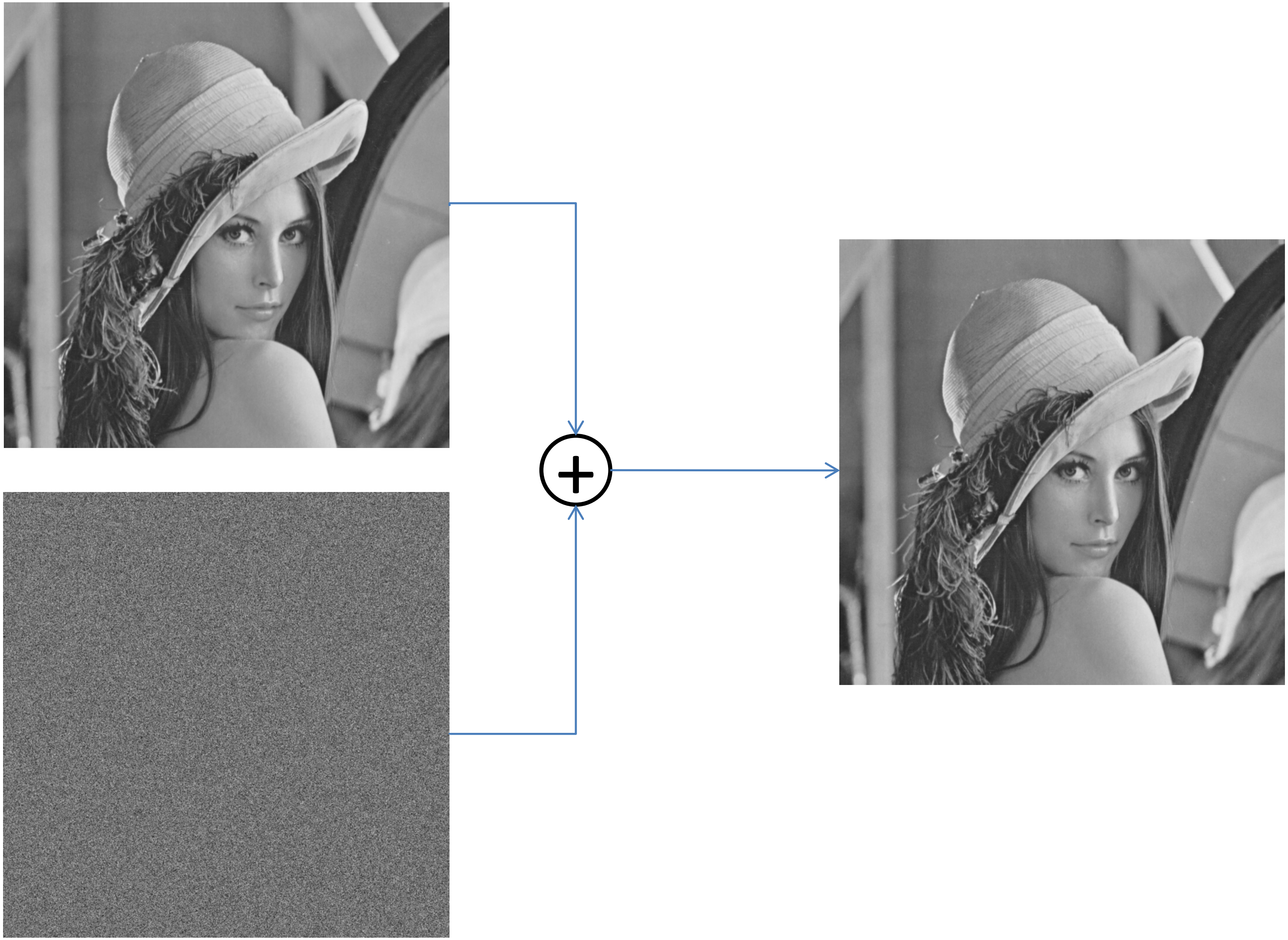
1. Calculate linear correlation z_{lc}
2. Detect message according to z_{lc}

$$z_{lc}(\mathbf{c_w}, \mathbf{w}_m) = \frac{1}{N}(\mathbf{c_o} \cdot \mathbf{w}_m + \mathbf{w_a} \cdot \mathbf{w}_m),$$

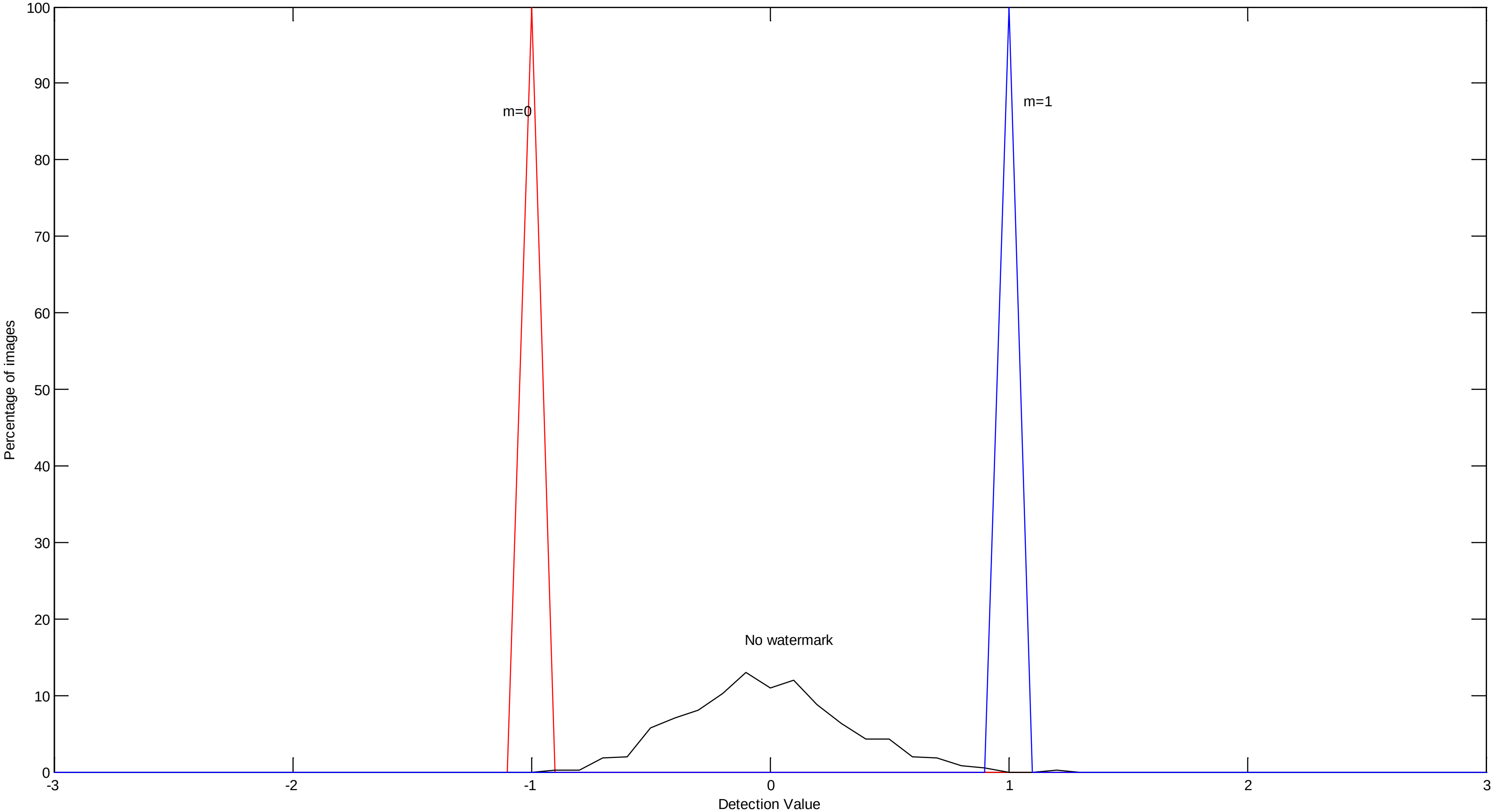
$$\alpha = \frac{N(\tau_{lc} + \beta) - \mathbf{c_o} \cdot \mathbf{w}_m}{\mathbf{w}_m \cdot \mathbf{w}_m}.$$

Geometric Interpretation

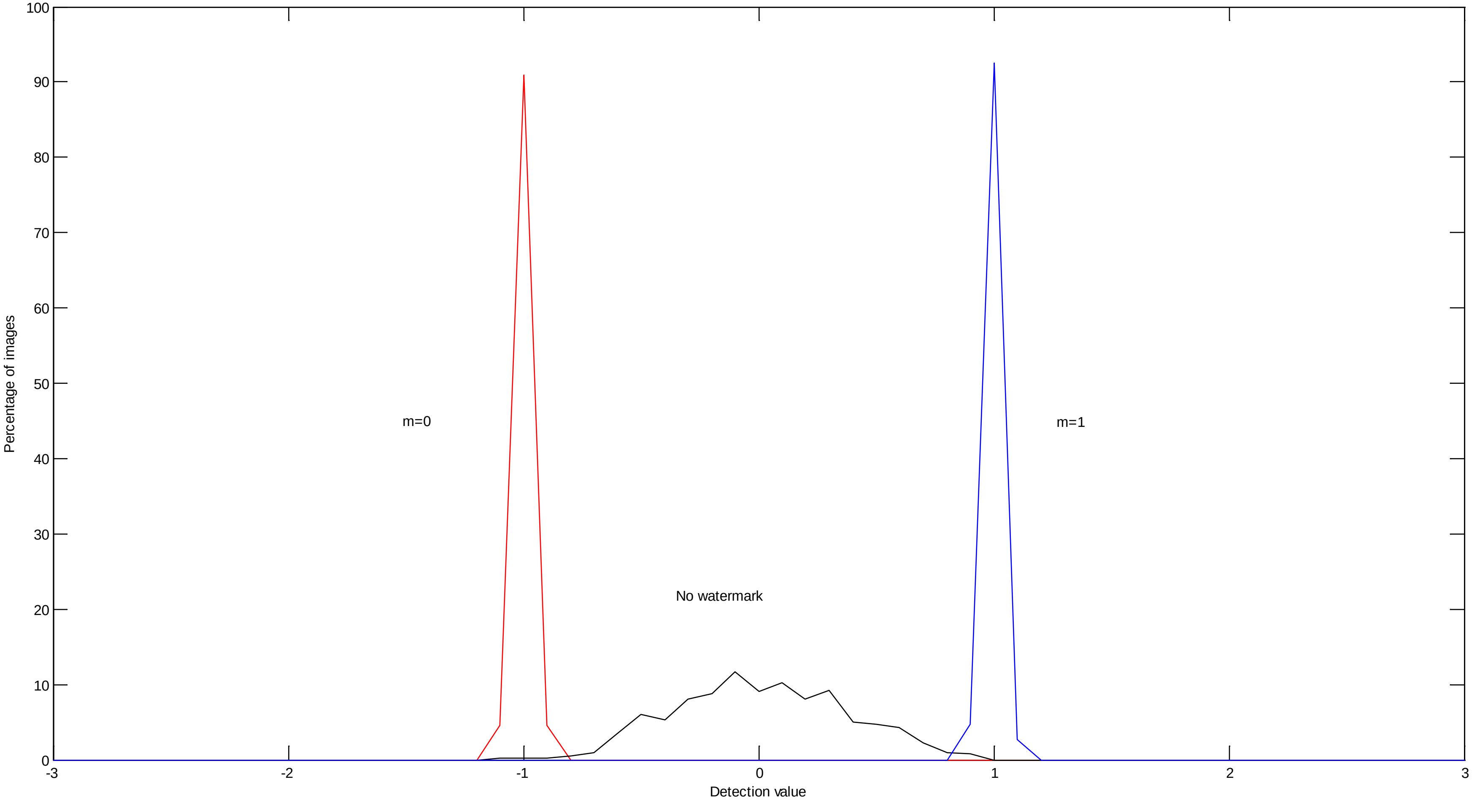




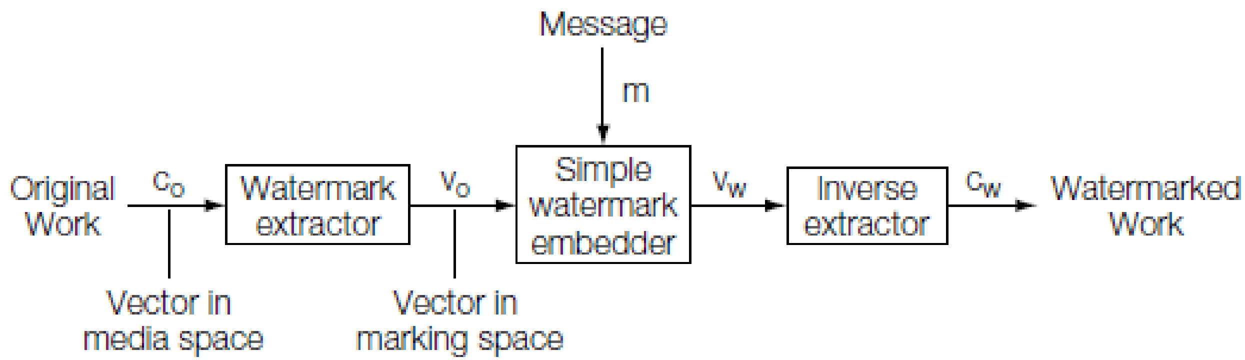
Effectiveness

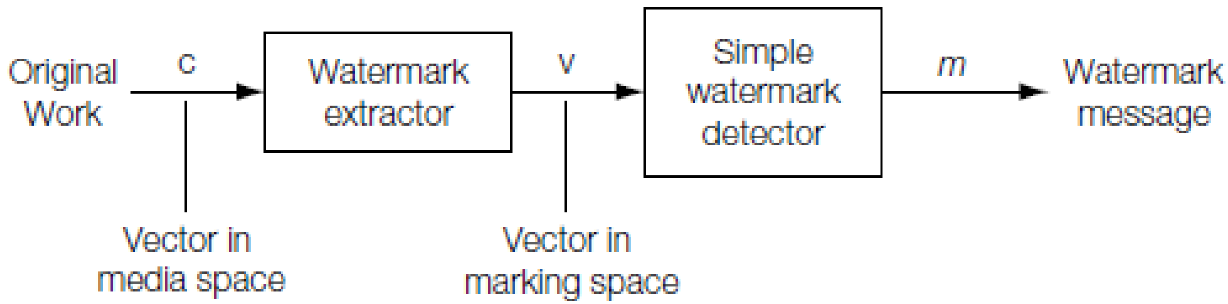


Adding Noise



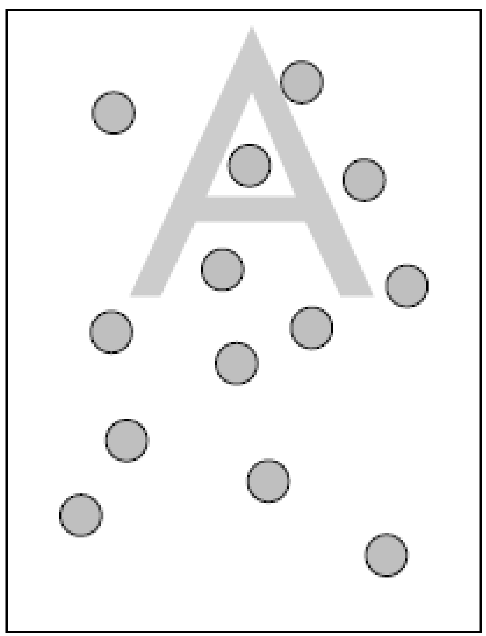
Exploiting Marking Space



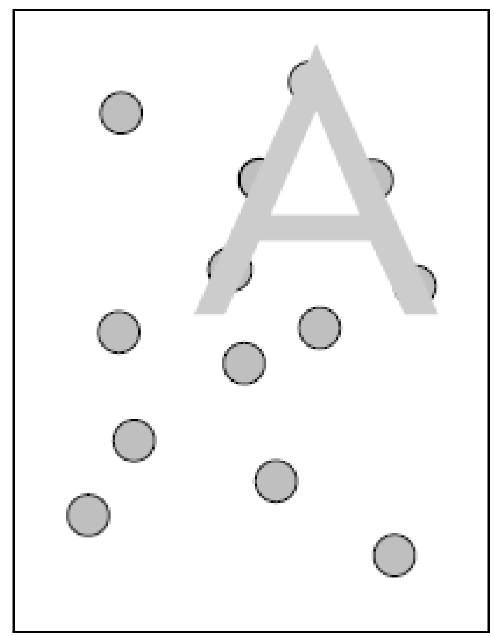


Dirty Paper Codes

- One code book comprised of subcode books for each message
- Select the code most similar to the original work



Blind writing



Informed writing

Block-Based/Fixed Robustness Embedding – Correlation Coefficient Detection

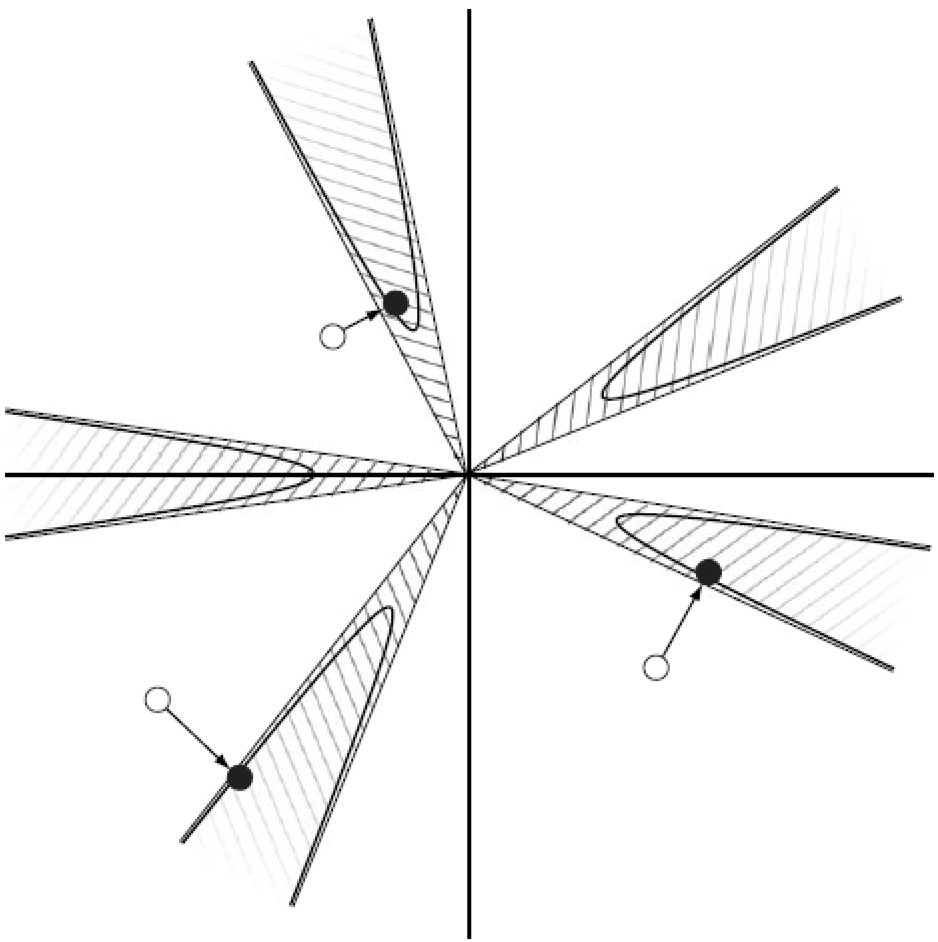
Embedder

1. Extract a watermark vector v_o by summing 8 x 8 blocks
2. Find the highest correlation between v_o and a set of reference marks (one set for 1, one set for 0)
3. Embed the highest correlation mark into the image using a fixed robustness algorithm

Detector

1. Extract a watermark vector v_o by summing 8 x 8 blocks
2. Find the highest correlation between v_o and the two sets of reference marks
3. If it's above the threshold then the message is detected

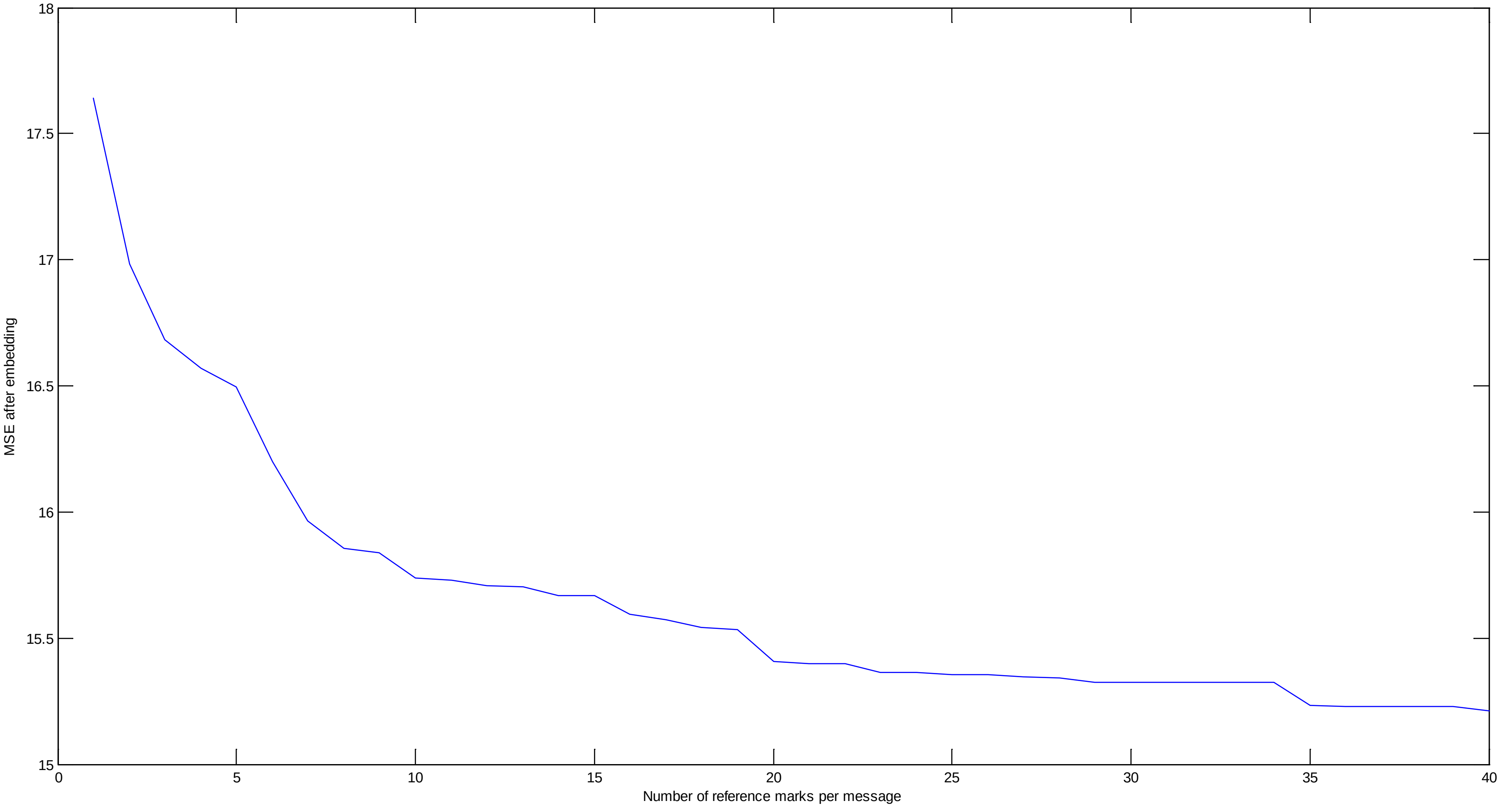
Geometric Interpretation



Before embedding

After embedding

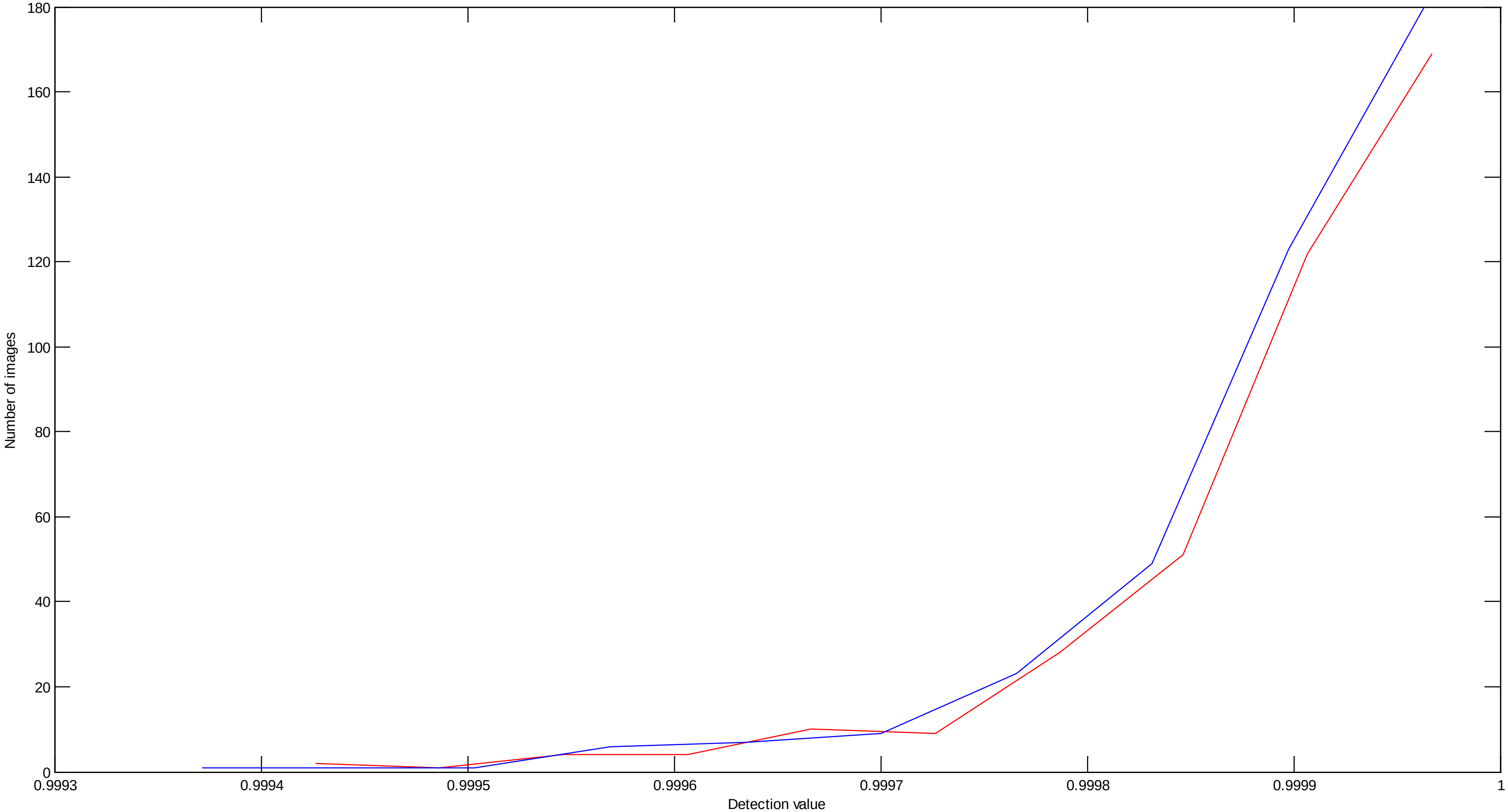
Fidelity



18
17.5
17
16.5
16
15.5
15
MSE after embedding
0
5
10
15
20
25
30
35
40
Number of reference marks per message

Effectiveness

- t_{nc}=0.95 R^2=30



Orthogonal Lattice Dirty Paper Code

Embedder

1. Encode the message into a sequence of coded bits using Trellis coding
2. Divide the image into 8 x 8 blocks
3. Modify each block to embed a bit using the reference mark

Detector

1. Compute correlation of each block with the reference mark and use it to find z

   $z = \text{floor}(\text{corr} / \alpha + 0.5)$
2. If z is odd then we have a 1, else we have a 0
3. Decode the message using the Viterbi decoder

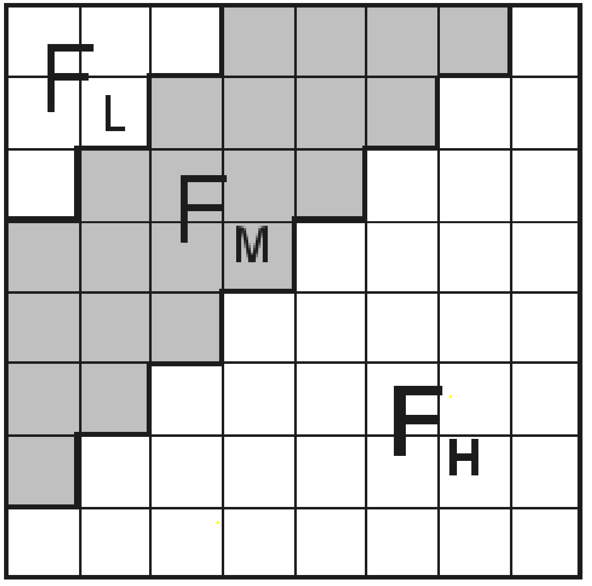
Before embedding

After embedding

Original message: 1024 bits
Embedded message: 4096 bits
MSE = 1.6927
Errors = 0

Other Methods

- Frequency Domain Based
  - Using DCT Coefficients
  - Using Wavelets



Thank You!